UNIVERSITÄT
BERN

TA Session 4

Anthony Gillioz
Institute of Computer Science – University of Bern, Switzerland

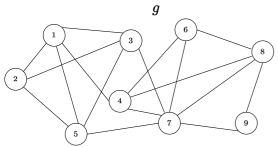
Contact: anthony.gillioz@unibe.ch

UNIVERSITÄT BERN

Theoretical Tasks

Task 1

1. Given the following graph g and three graph clusterings. Compute the graph density, the cut size (e.g., $cut(A, B)$), the intra-cluster density for each partition (e.g., $\delta_{int}(A)$), the general intra-cluster induced by the clustering (e.g., $\delta_{int}(g|A, B)$), and the inter-cluster density (e.g., $\delta_{ext}(g|A, B)$). After computing these metrics, you should analyze the values obtained and make an argument for which of the proposed clusterings is preferred.



1) $A = \{1, 2, 3, 4\}$ $B = \{5, 6, 7, 8, 9\}$
2) $C = \{1, 2, 3, 5\}$ $D = \{4, 6, 7, 8, 9\}$
3) $E = \{1, 2, 3, 6, 8\}$ $F = \{4, 5, 7, 9\}$

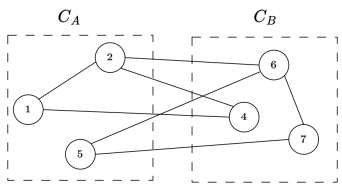
- Write down all the required metrics.
- Explain which clustering method should be favored based on the computed metrics.

UNIVERSITÄT BERN

Theoretical Tasks

Task 2

2. Which node pair would be changed in the following bisection in the Kernighan-Lin algorithm? Apply one turn of the Kernighan-Lin algorithm and provide the values E_v, I_v, and D_v for all nodes v in C_A and C_B as well as all values $R_{a,b}$ for all pair of nodes.



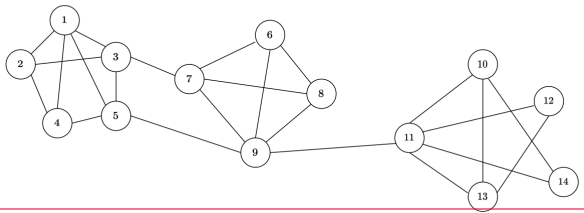
- Identify the node replacement that results in the most significant reduction in cut size and compute the cut size after the node substitution.

UNIVERSITÄT BERN

Theoretical Tasks

Task 3

3. Compute a spectral graph bisection (see Alg. 10 in the lecture notes) according to the Fiedler vector of the following graph. Apply the spectral bisection algorithm twice, with a threshold of $\theta = 0$, to identify three clusters. First, apply the algorithm to the entire graph to determine the initial partitioning. After obtaining this partitioning, apply the spectral bisection algorithm once again to the larger of the two clusters. For each step report the clustering and the Fiedler vector you obtain.



- Apply the spectral graph clustering on the complete graph Laplacian.
- Use the Fiedler vector to partition the graph.
- Apply a second time the clustering algorithm on the largest cluster previously obtained.
- Report the two Fiedler vectors and the three clusters you obtained.

Implementation Task

In this implementation task, you have to implement Prim's Minimum Spanning Tree (MST) algorithm (see Alg. 12 in the lecture notes).

Remarks:

- The entire code must be contained within the file `PR_lecture/Exercise_4/ex4.py`.
- You are allowed to modify the code as much as you want, including changing function signatures, creating new functions or classes, and so on.

Implementation Task
Remarks

Once you have implemented the MST algorithm apply it on the weighted graph in `PR_lecture/Exercise_4/graphs/graph_00.graphml` and save the spanning tree (i.e., the list of covered edges) as a list of tuples in `PR_lecture/Exercise_4/results/spanning_tree.txt`.

UNIVERSITÄT BERN

Implementation Task

Idea of code structure

```python
from utils import load_all_graphs
import networkx as nx


def PMST(graph: nx.Graph):
    # Code here

def ex4():
    graphs = load_all_graphs('./graphs')

    PMST(graphs[-1])


if __name__ == '__main__':
    ex4()
```